EXHIBIT 79

REDACTED

Eng Methodology for DFP system limits

October 21st, 2014

[redacted]

Why eng decided to implement limits

We always knew that we needed limits because they are a core component of a sound stability strategy for any world-class system.

If we could have waved a magic wand and had limits in place years ago, we would.

Some might view limits as meaning that Google can't scale. The truth is the opposite. The fact that we didn't have limits in the past is an indication that we didn't have scalability in mind as much as we should have.

Process used to determine limits

Eng considered three aspects:

1. How big could an individual publisher get before their network is at risk?
2. In light of DFP being a shared resource, how much load can the overall system take, and what limit would mitigate the risk of a pub affecting other pubs?
3. What is the current usage across the system?

To determine the answers to these questions, sometimes we were able to do a test and get a definitive number, other times we had to estimate using best-known information. For example, we were "lucky enough that there were some objects way over any reasonable limit we might pick already in existence. By seeing that there's no way to load them in the UI, it became clear that the limit needs to be smaller than those objects currently are.

Each of these three aspects could lead to a different limit calculation. We used judgement in balancing the answers to these three questions.

[redacted]

As an example, if a particular limit were to be breached, it could have an adverse effect on the extractor, which may result in delayed push times (or, worst case, failing pushes) to serving for all customers on a given shard.

We tended to err on the side of being conservative for a few reasons:

1. Once we say what the limit is, we know that will affect publisher behavior (as in they might expect to just brush up against the limit without any issues ever).
2. It's far more palatable to raise limits later than be forced to lower them.

Eng response to publisher issues around stated limits

Until now, there have been two instances where eng changed a limit in response to feedback from publishers.

Custom keys

We had initially proposed a limit of [redacted] values per free-form custom key. There are not many keys out in the system that go over this number. However, the ones that do cover a very important use case that was not apparent during analysis. Many publishers use zip-code targeting using a custom key. There are O(50k) zip codes in the US, which is obviously bigger than [redacted].

We were fortunate here because the [redacted] number was driven by IM/F scaling considerations. All other systems could support a bigger limit, in theory. [redacted] took on a major refactoring project in Q4 2014 which we expect will allow us to safely offer a [redacted] limit.

Targeting cardinality

Eng had proposed and implemented a limit of [redacted] targeting nodes per LI. There were two pieces of feedback:

1. It's hard for a human to count the number of targeting nodes
2. Many publishers have valid use cases where they need a much higher limit. Some made cases for up to [redacted] the limit.

[redacted]. There's [redacted] publishers getting a higher limit. The highest we are able to give with current infrastructure is a limit of 5k leaf targeting nodes.

You'll also note the use of the word "leaf." In response to the feedback, eng changed the counting methodology to only count leaf nodes in the targeting expression. This makes the limit slightly higher, but, more importantly, much easier to understand for a user.

Active line items

There is a serving-driven limit of a max of 61k active line items in a network. There are some publishers, [redacted], that expect to need a lot more active ads (~150k range).

Eng showed willingness to work with [redacted] on this. We don't have their data, since it only exists in [redacted]. The team proposed to have [redacted] import their data into a dummy network so that we could analyze it to see if [redacted] specific usage could be supported with a higher limit.

Known infrastructure improvements that are coming

Agave

DFP is migrating its data storage to F1. F1 is a highly-scalable non-relational data store with a relational-like interface (making it a viable replacement for a SQL database).

Eng is expecting great performance from the new storage system, but we do not yet have realistic performance numbers. It would definitely be counting the chickens before they come home to roost to make promises to publishers about what kind of performance improvement we may achieve.

One example of something that could go wrong is that, while the new system is theoretically faster, it might not be faster at the get-go because we haven't optimized our usage of it. Whereas we have gone to great lengths to optimize our usage of the MySQL ads db.

Another example of something that could go wrong is that the migration depends on a **lot** of other projects completing ([redacted]). And so there is a high variability in the expected completion date.

In terms of what could go well, for instance, we might get some improvements before fully migrating to Agave: as long as readers are reading from F1, they might see a performance improvement even if the writers are still on ads db.

Agave could have a positive impact on various limits, but we won't know which ones and how much for sure until we've moved and had time to measure the impact.

Competition limits within ad serving

Serving is considering limiting the number of line items that compete on a given request to a reasonably large number (like 1000). The line items would be randomly selected, and so over many requests the serving behavior shouldn't change. However, this means that an individual request can execute faster.

As a net result this could have a positive impact on the limit for the number of active line items.

Features that are coming

[redacted]

[redacted]

Features that publishers aren't using in the best way possible

[redacted]

Who decided the limits?

Eng overall is responsible. For the most part, [redacted] has led the effort and followed up internally with other eng teams. He's probably the best person to ask about this stuff going forward.